Experiment -5

Student Name: Shreya Vakil **UID:22BCS15135**

Branch: BE-CSE **Section/Group:22BCS_IOT-619/A**

Semester: 6th **Date of Performance: 25/02/2025**

Subject Name: Java

Subject Code: 22CSP-351

PROBLEM – 1

1. **CODE:**

```
import java.util.ArrayList;
import java.util.List;

public class AutoboxingUnboxing {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>();

        numbers.add(10);
        numbers.add(20);
        numbers.add(30);

        int sum = 0;
        for (Integer num : numbers) {
            sum += num;
        }

        System.out.println("Sum: " + sum);
```

```
        String numStr = "50";
        Integer parsedNum = Integer.parseInt(numStr);
        System.out.println("Parsed Number: " + parsedNum);
    }
```

1. **OUTPUT:**

PROBLEM2:

1. Code:

```
import java.io.*;

class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    int id;
    String name;
    double gpa;

    public Student(int id, String name, double gpa) {
        this.id = id;
        this.name = name;
        this.gpa = gpa;
    }

    public void display() {
        System.out.println("ID: " + id + ", Name: " + name + ", GPA: " + gpa);
    }
}

public class StudentSerialization {
    public static void main(String[] args) {
        Student student = new Student(101, "John Doe", 3.8);
        String filename = "student.ser";

        // Serialization
        try (ObjectOutputStream out = new ObjectOutputStream(new
    FileOutputStream(filename))) {
            out.writeObject(student);
            System.out.println("Student object serialized.");
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }

        // Deserialization
        try (ObjectInputStream in = new ObjectInputStream(new
    FileInputStream(filename))) {
            Student deserializedStudent = (Student) in.readObject();
            System.out.println("Deserialized Student Details:");
            deserializedStudent.display();
        } catch (FileNotFoundException e) {
            System.out.println("File not found.");
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

2. OUTPUT:

```
Student object serialized.
Deserialized Student Details:
ID: 101, Name: John Doe, GPA: 3.8
```

3. LEARNING OUTCOMES:

- Understand Serialization and Deserialization – Learn how to convert Java objects into a byte stream and restore them.
- Use of ObjectOutputStream and ObjectInputStream – Gain experience with Java's built-in serialization mechanisms.
- File Handling in Java – Learn how to read and write objects to files.
- Exception Handling – Effectively handle FileNotFoundException, IOException, and ClassNotFoundException.
- Practical Application – Implement object persistence for real-world scenarios